

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
25 October 2001 (25.10.2001)

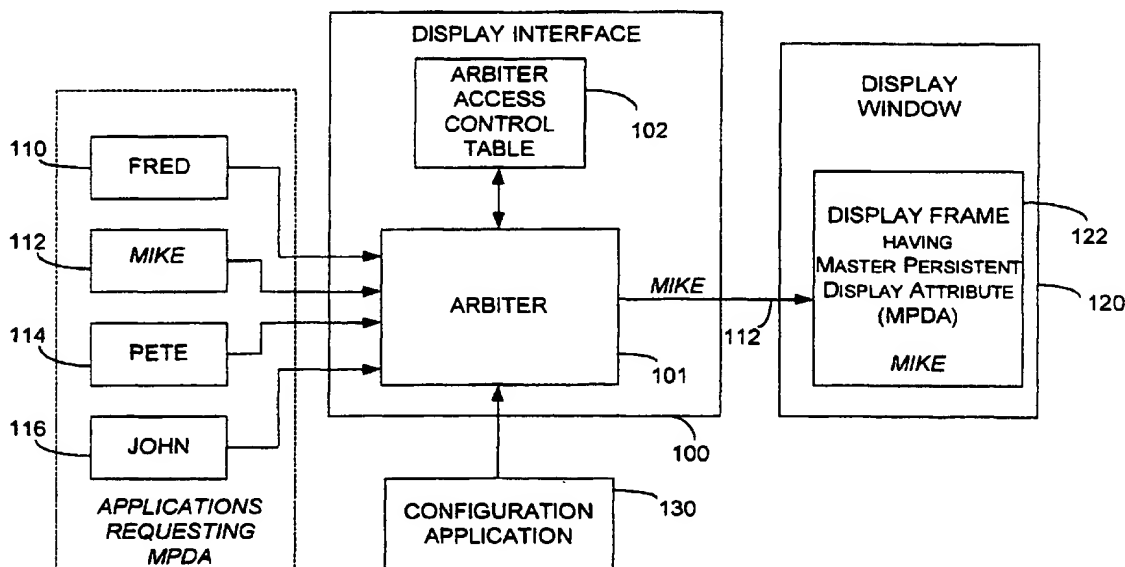
PCT

(10) International Publication Number
WO 01/80215 A1

- (51) International Patent Classification⁷: **G09G 5/14**, G06F 9/46
- (74) Agent: **O'ROURKE, John, F.**; Christie, Parker & Hale, LLP, Post Office Box 7068, Pasadena, CA 91109-7608 (US).
- (21) International Application Number: PCT/US01/12795
- (22) International Filing Date: 19 April 2001 (19.04.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/198,315 19 April 2000 (19.04.2000) US
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (71) Applicant (*for all designated States except US*): **BROAD-COM CORPORATION** [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (*for US only*): **DOVE, Michael** [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US).
- Declaration under Rule 4.17:**
— *of inventorship (Rule 4.17(iv)) for US only*
- Published:**
— *with international search report*

[Continued on next page]

(54) Title: APPARATUS AND METHOD FOR PERSISTENT DISPLAY INTERFACE



(57) Abstract: An apparatus and method for maintaining a graphics display window on a display screen that will not be obscured by any other graphics display window. The graphic display apparatus comprises an arbiter, a gatekeeper and an access control table. The arbiter selects from among a plurality of computer applications requesting authorization on the display screen to display data. Authorization is granted via a persistence attribute. Once a persistence attributed is obtained, the graphics display window remains on the screen unobscured.

WO 01/80215

PCT/US01/12795

WO 01/80215 A1



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

1 APPARATUS AND METHOD FOR PERSISTENT DISPLAY INTERFACE

CROSS-REFERENCE TO RELATED APPLICATION(S)

5 This application claims priority of provisional patent application No. 60/198,315 filed April 19, 2000, and incorporates said application by reference in its entirety.

BACKGROUND OF THE INVENTION

10 Modern computers employ human-computer interfaces (HCI) which enable users, including system administrators, programmers, and end users, to interact with one or more computer systems via an easy-to-use, visually-oriented display, typically on the screen of a computer monitor. The screen may have information divided into, and distributed among, one or more conceptual frames, called windows, through which the user manages, for example, data, files, system programs, and application programs.

15 Window-oriented operating systems (OS) are ubiquitous to the extent that nearly all contemporary computer systems implement one or more such systems. Such operating systems can include, without limitation, the open system architecture X-Windows which is supported worldwide by several hundred vendors and original equipment manufacturers (OEM); Mac-OS® by Apple Computer, Inc., Cupertino, CA; and the WINDOWS® family of operating systems from Microsoft Corp., Redmond, WA.

20 In general, each of the one or more overlapping, often rectangular, windows acts like a separate terminal, and the user is enable to perform different tasks in each window. The computer operating system employs one form of HCI, namely, a graphical user interface (GUI) to permit the user to take advantage of the computer's graphics capabilities to make a program easy to use. Indeed, a well-designed GUI manages windows in a manner that frees the user from learning complex command languages, and allows the user to interact with the data, files, system, and programs in an intuitive manner.

25 Windows have geometry as well as delimited regions, icons and different visual attributes associated with them. Display attributes typically are manipulated. The OS API at the lowest level often provides the interface with the pixels on the screen. At a higher level, a user interface API can build borders, insert title bars and stands, etc.

30 For certain applications, it is desirable to maintain a specific graphics display window on a screen that will not be obscured by any other window. This is typically accomplished through the Application Program Interface to the window. An "always on top" (AOT) attribute maintains a specific graphics display window on the screen, unobscured by any other window. Initially, a particular application program can seize and retain this attribute so long as no other application program requests it. A subsequently executed application program can take control of this

35

1 attribute, and itself use the AOT feature. Clearly, this attribute is not in a persistent state, because it belongs to an application program so long as no other application program seizes control of it.

Some application programs try to implement a persistent, AOT attribute, by implementing a re-entrant loop, which is constantly setting the AOT attribute. If another application program
5 attempts to seize the AOT attribute, the first program re-enters the loop and regains ownership of the attribute.

Another example of attempting to create a persistent display attribute, including an "Always, Always on Top" (AAOT) attribute which may be found in the case of a video overlay on a television card for a computer. In this example, the image bits are written directly into the
10 display buffer. The operating system is generally unaware of what graphics display data is written to that section of the buffer, only knowing that the corresponding memory segment is reserved. Thus, the OS is undesirably prevented from having control over the data bits in the image display buffer. Another method of implementing an AOT attribute is to reserve portions of the screen buffer, to the detriment of the operating system. The OS is unable to take advantage
15 of these reserved portions of the screen buffer because they are hidden from the OS.

Current attributes typically are assigned according to a most-recently-requested basis, or a first-come-first-serve basis. However, it is desirable to assign an attribute such that it always preempts other attributes.

20 SUMMARY OF THE INVENTION

The present invention satisfies the above needs by providing apparatus, method and computer program products which assigns a persistent attribute and, in particular, a master persistent display attribute, that is capable of preempting other attributes assigned or seized during the typical course of display operation. According to the invention, a graphics display
25 window can be maintained on a display screen without being obscured by any other window.

One inventive apparatus produces a perceptible representation of data, and includes an arbiter that selects a dominant program from among multiple programs seeking a master persistence attribute to display program data according to a predetermined priority technique, and assigns the master persistence attribute to the dominant program. In addition, the arbiter can be
30 coupled with an access control table which contains indicia representative of the predetermined priority scheme. Such indicia can include process ID (PID), window ID (WID), priority, revoked and repudiated credentials, authentication token or key, master persistence attribute authorization, descriptive text, program status, system status, an accessible display region, and an excluded display region. The arbiter can be configured with the predetermined priority scheme with a
35 configuration application program, coupled with the access control table. An I/O manager can be used to manage display between the program and the display, and can communicate the data

1 through an intervening graphics device driver. Moreover, the graphics device driver can be
coupled with a graphics display buffer. The arbiter can include a rules engine, a state machine,
and a content-addressable memory that provides the predetermined priority scheme for
determining dominant program priority. The apparatus also can be configured to include a
5 gatekeeper which determines selected ones of programs to be granted access to the arbiter to
receive the master persistence attribute according to a predetermined access scheme. The
gatekeeper can be coupled with a configuration table, which stores an indicia representative of
the predetermined access scheme. These indicia can include a process ID (PID), window ID
(WID), priority, revoked and repudiated credentials, authentication token or key, master
10 persistence attribute authorization, descriptive text, program status, system status, an accessible
display region, and an excluded display region. The apparatus also may be configured to employ
a gatekeeper alone, in which case the gatekeeper will determine the priority of programs seeking
an attribute and grant the attribute accordingly.

In another embodiment, the invention can include a graphics system, which includes a
15 video input receiving a graphical data signal; a video output coupled with a display; a display
controller coupled to the video input signal and selectively transmitting the graphical data signal
to the video output; and an arbiter coupled to the display controller, the arbiter effecting the
selectively transmitting by granting a persistence attribute according to a predetermined priority
scheme. The display controller can selectively transmit the graphical data signal, responsive to
20 the arbiter. This embodiment also can include a CPU interface which couples the graphics
system with a CPU. The CPU receives and transmits display control signals. The arbiter is
responsive to the display control signals transmitted by the CPU. The CPU can include a
gatekeeper, which is operably coupled with the arbiter and which transmit the predetermined
priority scheme thereto. Either, or both, of the arbiter and the gatekeeper can have a table
25 coupled respectively therewith, which stores indicia relevant to the respective access or priority
scheme. These indicia can include process ID (PID), window ID (WID), priority, revoked and
repudiated credentials, authentication token or key, master persistence attribute authorization,
descriptive text, program status, system status, an accessible display region, and an excluded
display region.

30 The invention also includes a method of assigning a persistence attribute to at least one
of a plurality of dominant programs, including requesting the master persistence attribute from
a gatekeeper; assigning a set of priority rules to the gatekeeper via a configuration application
program; the gatekeeper granting keys to selected dominant application programs allowing access
to an arbiter; the arbiter examining an arbiter access control table storing the predetermined
35 priority scheme; and the arbiter assigning the persistence attribute to the at least one of a plurality
of dominant application programs granting access to a display window.

1 Furthermore the invention can include a computer program product recorded on a computer readable medium that implements the inventive method herein as well as is functionally equivalent to the several apparatus described herein.

5 The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the following drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

10 These and other features, aspects and advantages of the present invention will be more fully understood when considered with respect to the following detailed description, appended claims and accompanying drawings, wherein:

FIG. 1 is illustrates a display interface according to the present invention using an arbiter;

FIG. 2 illustrates a display interface according to another embodiment using an arbiter and a gatekeeper;

15 FIG. 3 illustrates a software embodiment of the display interface invention in the context of an operating system;

FIG. 4 illustrates an embodiment of a display interface invention where an arbiter is implemented in graphics display hardware;

FIG. 5 illustrates an embodiment of the present invention where a gatekeeper is implemented in software and an arbiter is implemented in hardware;

20 FIG. 6 illustrates a software embodiment of the display interface invention in the context of the Microsoft WINDOWS® NT® 4.0 operating system;

FIG. 7 illustrates an embodiment of the present invention where a gatekeeper is implemented in software and an arbiter is implemented in hardware, in the context of the Microsoft WINDOWS® NT® 4.0 operating system;

25 FIG. 8 illustrates a graphics display screen in which selected portions of the screen are allocated according to aspects of the invention herein; and

FIG. 9 is a block diagram of an advanced graphics display system which implements aspects of the present invention.

30 DETAILED DESCRIPTION OF THE INVENTION

The present invention provides apparatus and methods that provide one or more master, persistent interface attributes (MPIA), which continually maintains a specific HCI attribute in a selectable state. Except for the operation of MPIA apparatus and methods therefor, the selectable state can not be altered, or superceded, by programs, routines, or code, which otherwise would
35 be capable of altering the state of the attribute.

For example, in the case of a visually-oriented HCI, such as a GUI, or window, the

1 persistent interface attribute could be a master, persistent display attribute (MPDA) such as, for
example, an "Always, Always on Top" (AAOT) MPDA that maintains a preselected graphics
display window, subwindow, or frame, (collectively, a "window") granted the attribute, at the
highest Z-order level, without being overwritten. As used herein, "Z-order" describes the
5 apparent visual depth of a window in a display. For example, a window or subwindow of a
higher Z-order can appear to overlay partially, or completely, a window, subwindow, frame, or
pane, of a lower Z-order. Also as used herein, a "window" can include predefined portions of
a display including, without limitation, title bars, icons, status monitor indicators, system trays,
and the like.

10 In some implementations, it may be desirable that the AAOT MPDA be implemented
through the code in the OS that deals with writing the pixels on the screen so that, if a particular
window is assigned that attribute, no other window, or subwindow, being rendered on the screen,
would have access to write to those pixel memory locations, which occurs when a window
overlaps. Preferred embodiments of the present invention implement arbitration, which can
15 assigns a MPDA, such as an MPDA, according to a predetermined priority technique. It is
desirable that the priority scheme of the present invention determine those programs which are
eligible to make use of the MPDA and, of those, which program which actually receives it.
Where a program has possession of the MPDA, the priority scheme and whether that program
should retain possession.

20 In one implementation, this functionality could be present in the OS, preferably at the
lowest level, where there can be some arbitration for access to the screen resources. Thus, it is
desirable that the OS arbitrate, for example, the pixels on the screen with the AAOT, or the "Z-
ordership, persistents display attribute, such that no other application program other than the
application program granted the MPDA, may access and overwrite those pixels.

25 One example of an application for such a priority scheme includes providing a persistent
pop-up display window for transmitting critical information from an emergency broadcasting
system in case of hurricane, earthquake, tornado or the like, whether the user display is supplied
via cable, telephonic, broadcast, or wireless services. Because it is contemplated that the present
invention have multimodal interface capability, the present invention is not limited to visually-
30 oriented interfaces, e.g., visual displays, but also can include textual, audio, and other forms of
perceivable information, and combinations thereof. However, for clarity, all such perceivable
representations will be called "displays" herein. Furthermore, although aspects of the invention
herein may be described in terms of a visually-oriented interface as implemented on a stand-alone
personal computer (PC), this limitation also is solely for the purpose of clarity and is an artifact
35 of presentation. Indeed, aspects of the present invention can be implemented on networked
computers, including servers, and in a myriad of communication devices, including without

1 limitation, handheld computing devices, wireless telephones, set-top boxes, cable modems, and
the like. In the context of the present invention, a "computer" can include, of course, a
present-day traditional, stand-alone computer, as well as any intelligent device capable of
receiving, transforming and producing a representation of data that is perceptible to one or more
5 senses. Also as used herein, the term "program" can include programs, processes, threads, and
the like, whether invoked by a user application, a system resource, a supervisory monitor, a
configuration program, and so forth. Moreover, any or all components of the present invention
can be implemented in hardware, software, or a combination thereof.

Figure 1 illustrates a display interface 100, according to the present invention. In many
10 operating systems, the ultimate ownership of a graphics display window is non-exclusively
granted to a program or process on a first-come, first-served basis. Because embodiments of the
present invention are preferred to implement arbitration, according to a predetermined priority
technique, arbiter 101 is provided to select from among computer programs 110, 112, 114 and
116 that are requesting an MPDA. Typically, ownership or control of a portion of a display is
15 non-exclusive to the extent that a second, later-executing program or process can acquire
ownership or control of a graphics display portion at the expense of a prior owner. This can
result in the first display window being overwritten or superseded in Z-order by the
subsequent program. Thus, arbiter 101 can determine which of programs 110, 112, 114, 116 will
be exclusively granted ownership of an MPDA and access to selected pixel memory locations,
20 which locations have predetermined portions of the graphical display corresponding therewith.
Also, windows and sub-windows typically overlap visually on a first-come, first-served, Z-order
priority basis. Arbiter 101 can assign an exclusive Z-order to a particular window, subwindow,
or frame.

Arbiter 101 also can be configured to determine where the programs and process granted
25 an MPDA may place graphical data, a display frame, or a subwindow on a display. The MPDA
could be assigned to the graphics display window, e.g., subwindow 122, that is to be maintained
unobscured by any other window on display screen 120.

Arbiter 101 can include a very simple rules engine which allocates priority using a
predetermined rule set. Arbiter 101 receives MPDA access requests from programs 110, 112,
30 114, 116, and can interpret the requests to determine which program will be allowed to access
a particular region of display 120 and to receive the MPDA. If a priority conflict arises among
programs 110, 112, 114, 116, then arbiter 101 also can be configured to resolve the conflict using
the predetermined rule set. Arbiter 101 also may perform request authentication, and deny access
to programs 110, 112, 114, and 116, if the request is determined to be inauthentic.

35 In addition to, or in place of, the internal rule set, arbiter 101 can cooperate with arbiter
access control table 102 to establish the access priority relative to display subwindow 122 in

1 display window 120. Specifically, the arbiter access control table 102 can store information
regarding which program 110, 112, 114, 116 may be granted the MPDA. can include pertinent
credentials, including, without limitation, process ID (PID), window ID (WID), priority, revoked
and repudiated credentials, authentication token or key, MPDA authorization, descriptive text,
5 program status, system status, accessible and excluded display regions, and the like. A process
may open multiple windows, thus, it may be desirable to assign a MPDA to all windows in that
process, and, then, allow the process to further determine access by related subprocesses to
corresponding frames, panes, subwindows, and so forth in the windows accessed by that process.
When arbiter access control table 102 is employed, arbiter 101 may be configured to draw solely
10 upon the information in look-up table 102 to decide which of programs 110, 112, 114, and 116,
will receive access to the MPDA and, if granted, the priority of such access. Alternatively,
arbiter 101 can be configured to cooperate with table 102 to dynamically allocate and re-allocate
an MPDA among one or more programs or processes.

FIG. 2 illustrates another embodiment of a display interface 200 according to the present
15 invention, in which gatekeeper 204 is used to decide whether access to arbiter 201 is granted or
denied. Display interface 200 also can include arbiter access control table 202, which can be
similar to arbiter access control table 102 in FIG. 1. Display interface 200 also may employ
gatekeeper configuration table 206 to store application-specific access information for use by
gatekeeper 204. In one embodiment of the invention herein, gatekeeper 204 can receive a request
20 for an MPDA from programs 210, 212, 214 and 216, and select which of programs 210, 212,
214, 216, will be permitted to access arbiter 201 with an MPDA request. The selection may be
based upon configuration parameters stored in gatekeeper configuration table 206, which can be
provided by way of configuration application 230. Arbiter access table 202 can contain
parameters specific to those programs, or classes of programs, that will be permitted to request
25 an MPDA. In addition, arbiter 201 also may be programmed with one or more predetermined
rules governing granting of an MPDA. Based on the information stored in table 202, or the
predetermined rules, arbiter 201 can allow a selected program 210, 212, 214, 216 to employ an
MPDA to create display frame 222, within display window 220.

In one embodiment of the present invention, gatekeeper 204 can accept requests from
30 programs 210, 212, 214, 216, which may request an MPDA and, if the requestor is entitled to
access arbiter 201, gatekeeper 204 issues a corresponding key or token, allowing access to arbiter
201. In one implementation, gatekeeper 204 can selectively grant access by programs 210, 212,
214, 216 to arbiter 201. In addition, gatekeeper 204 can selectively assign priority to programs
210, 212, 214, 216, as well as manage authentication data regarding programs which gatekeeper
35 204 may encounter during initialization or runtime. In the latter case, gatekeeper 204 further can
be configured to provide each of selected ones of programs 210, 212, 214, 216, with a key or

1 token, to present to arbiter 201 at a specified time during program execution. Configuration
control table 206 can be used to store and communicate information, including the
aforementioned program credentials, about classes of programs as well as specific programs,
which gatekeeper 204 may encounter. This information can include can include pertinent
5 credentials, including, without limitation, process ID (PID), window ID (WID), priority, revoked
and repudiated credentials, authentication token or key, MPDA authorization, descriptive text,
program status, system status, accessible and excluded display regions, and the like.

When arbiter 201 receives a key from one of programs 210, 212, 214, 216, it can then
determine whether the key is authentic and, if it is, assign an MPDA to a selected program,
10 granting access to display 220, or a predetermined portion thereof, for example, display frame
222. Arbiter 201 also can cooperate with arbiter access control table 202 to store and manage
information relating to, for example, programs authorized to access arbiter 201. This information
also can include pertinent credentials, including, without limitation, process ID (PID), window
ID (WID), priority, revoked and repudiated credentials, authentication token or key, MPDA
15 authorization, descriptive text, program status, system status, accessible and excluded display
regions, and the like.

In one embodiment of the invention herein, both arbiter 201 and gatekeeper 204 can be
realized completely in software, for example, in the operating system kernel. In other
embodiments, the functions of arbiter 201 and gatekeeper 204 can be distributed in both software
20 and hardware, whether the functions are resident in a particular device, or distributed across
multiple devices. Although configuration by an end user may be desirable under some
circumstances, it is typically preferred that gatekeeper 204 and, thus, arbiter 201, be configured
by a network administrator, original equipment manufacturer (OEM), value-added retailer
(VAR), system provider, or other authorized user, so that the capability to use an MPDA is
25 restricted to beneficial and intentional uses. Thus, it may be desirable to realize gatekeeper 204,
as well as arbiter 201, as a protected kernel level function. Access to gatekeeper 204, gatekeeper
configuration table 206, arbiter 201, and arbiter access control table 202, may be implemented
via a concealed user-level function, for example, an undocumented Application Program
Interface (API), or, where the Microsoft Windows® OS is used, via the system registry
30 (WINDOWS® is a trademark of Microsoft Corp., Redmond, WA), although other known access-
control techniques may be used. Gatekeeper 204 can provide housekeeping services to arbiter
201 by dynamically indicating changes of program status, accessible and excluded display
regions, revoked and repudiated credentials, and the like.

In yet another embodiment of the invention, gatekeeper 204 can be operable on one
35 computer physically distinct from the computer on which arbiter 201 is operable. Therefore, the
invention herein also contemplates being implemented in an environment of networked wired

1 and wireless intelligent devices.

The aforementioned principles are now illustrated by example, using FIG. 2. In one example, upon initialization, program "FRED" 210, program "MIKE" 212, program "PETE" 214, and program "JOHN" 216, can communicate with gatekeeper 204 to request access to, and
5 authorization to later use the MPDA attribute by, arbiter 201 during program run time. Based on internal rules, or with information stored and managed in gatekeeper configuration table 206, or both, gatekeeper 204 access to arbiter 201 can be granted to program "MIKE" 212 and program "JOHN" 216. In turn, based on internal rules, information stored and managed in access control table 202, or both, arbiter 201 can select program "MIKE" 212 to access display window 220,
10 granting exclusive control over display frame 222 to program "MIKE" 212, and denying access by program "JOHN" 216.

In an alternative embodiment, also illustrated by FIG. 2, program "FRED" 210, program "MIKE" 212, program "PETE" 214, and program "JOHN" 216, can communicate with gatekeeper 204 to request future access to arbiter 201. This time, program "FRED" 210, program
15 "MIKE" 212, and program "JOHN" 216, each are determined by gatekeeper 204 to be allowed access to arbiter 201. Each are provided with credentials, including perhaps a key, that will permit access to arbiter 201. Some, or all, of the information constituting the credentials for a particular program may be maintained by the respective program, or may be retained in part by gatekeeper 204, arbiter 201, or both. If program "PETE" 214 is not authorized to access display
20 window 220, gatekeeper 204 denies program 214 access to arbiter 201. Assume now arbiter 201 communicates with gatekeeper 204 that, until further notice, it can only allow access by two additional programs. Also assume that, for the purposes of the present example, program "MIKE" 212 and program "JOHN" 216 were granted higher priorities than program "FRED" 210. When program "FRED" 210, program "MIKE" 212, and program "JOHN" 216
25 communicate corresponding credentials to gatekeeper 204, seeking access to arbiter 201, only program "MIKE" 212, and program "JOHN" 216 are granted access. Access to arbiter 201 by program "FRED" 210 may be deferred until arbiter 201 is able to accept access by program 210, or may be cancelled altogether.

Figure 3 illustrates yet another embodiment of a display interface according to the present
30 invention, this time in the context of operating system 300. Similar to the embodiments described relative to FIG. 1 and FIG. 2, the embodiment illustrated in FIG. 3 includes arbiter 301. Similar to FIG. 2, the embodiment illustrated in FIG. 3 can include arbiter access control table 302, gatekeeper 304, and gatekeeper configuration table 306. Both tables 302 and 306 may include pertinent credentials, including, without limitation, process ID (PID), window ID (WID),
35 revoked and repudiated credentials, authentication token or key, MPDA authorization, priority, descriptive text, program status, system status, display status, accessible and excluded display

1 regions, and the like. The actual information maintained in either table can depend upon the
functionality desired of arbiter 301 and gatekeeper 304.

5 In this embodiment of the invention, gatekeeper 304 and table 306 can be maintained at
the user level of operating system 300, and arbiter 301 and table 302 can be maintained at the
more secure kernel level of the operating system 300. Arbiter 301 communicates with graphics
device driver 303 which, in turn, writes graphical data to display buffer 340 from those programs
310, 312, 314 authorized to do so by arbiter 301. The data written into hardware buffer 340 is
depicted on display 320, in the manner suited for display hardware 325.

10 During initialization, selected programs 310, 312, 314 can request authorization to create
an exclusive, persistent window on display 320. Where used, gatekeeper 304 can process a
request by programs 310, 312, 314 to create a graphical entity on display 320, which may possess
special attributes, including, for example, a master persistent display attribute, such as the AAOT
MPDA. Application programs 310, 312 and 314 can communicate with gatekeeper 304,
requesting authorization to access arbiter 301 which, in turn, can authorize access to user display
15 320 using, for example, a MPDA. Programs 310, 312, 314 can provide credentials to gatekeeper
304 as part of the authorization process. Advantageously, gatekeeper can employ gatekeeper
configuration table 306 to store and manage program credentials as well as additional credentials,
including keys, issued by arbiter 301 and gatekeeper 304. Responsive to the credentials
provided, gatekeeper 304 may grant, or deny access to arbiter 301 by programs 310, 312, 314.
20 The response of gatekeeper 304 to programs 310, 312, 314 can be based upon simple rules
programmed into gatekeeper 304, or in the alternative, information that is stored in an gatekeeper
configuration table 306. Where gatekeeper 304 is not desired, programs 310, 312, 314 can
provide credentials, keys, or tokens, directly to arbiter 301.

25 Selected programs 310, 312, 314 may then access arbiter 301 immediately, or receive an
authenticating token or security key from gatekeeper 304 for later use during the program's run
time. When accessed, arbiter 301 can determine which of programs 210, 312, 314 is permitted
to use a MPDA, and in which specified region of user display 320 particular program 310, 312,
314 is allowed to provide its display/information. This determination can be based upon simple
rules programmed into arbiter 301 or, in the alternative, upon information that is stored in an
30 arbiter access control table 302. Furthermore, either or both of arbiter 301 and gatekeeper 304
can be selectively controlled and supervised by one or more software agents. Agents are software
programs that are capable of autonomous, flexible, purposeful and reasoning action in pursuit
of one or more goals. Such agents can operate within a single computer environment, or across
a disseminated interwork of computers.

35 Figure 4 illustrates yet another embodiment of a display interface according to the present
invention. A display interface having arbiter 401 is implemented in display hardware 425. In this

1 configuration, a gatekeeper is not implemented and arbiter 401 alone selects among application
programs 410, 412, 414 which will be permitted to express the MPDA on user display 420.
Arbiter 401 can receive application program access parameters from OEM configuration
application 430 via I/O manager 407. I/O manager 407 can be located in solely software, and can
5 control all system input and output. I/O manager 407 also can be implemented in hardware, or
firmware. The implementation in FIG. 4 can be suitable where a general purpose operating
system is not used, and it is desirable to use a minimalistic program, rules engine, or state
machine to implement the arbiter access functionality. These parameters may be used by arbiter
401 to determine which application program, 410, 412 or 414, will have access to a specific
10 display region of user display 420. The access determination by arbiter 401 may either be by one
or more simple rules, or by reference to arbiter access control table 402. Table 402 can include
such information as a security token or authentication key, a process ID (PID), window ID (WID)
and, if desired, priority. Once an application program has been granted a MPDA by arbiter 401,
it accesses graphic display device driver 403 which in turn writes the application program's
15 graphic data to display buffer 440. The application program's display data in display buffer 440
then is made available on user display 420.

In Figure 5, still another embodiment of the display interface is presented. In this
configuration, however, gatekeeper 504, and corresponding configuration table 506, are
maintained in software and operating system 500, preferably at user level 516. Also, arbiter 501
20 and its corresponding arbiter access control table 502 are implemented directly in display
hardware 525. Application programs 510, 512 and 514 communicate with gatekeeper 504
requesting authorization to access arbiter 501 which can permit access to user display 520 using
a MPDA. Gatekeeper 504 may grant, or deny, the application programs access to arbiter 501.

During the normal operation of operating system 500, application manager 509 made
25 manage requests for resources from application programs 510, 512, 514, and negotiate with
system executive 505 for those resources. Typically, when an application program has need to
display information on user display 520, the system executive 505 direct the output request to
I/O manager 507 which, in turn, transmits the information to graphics device driver 503.
Graphics device driver 503 then inputs the information into display buffer 540, allowing the
30 information, now in suitable format, to be read and displayed by user display 520. In accordance
with the present invention, arbiter 501 communicates with, and controls, graphics device driver
503 such that only preselected ones of application programs 510, 512, 514, are permitted to
access the graphics device driver 503 with a MPDA. Gatekeeper 504 can communicate with
arbiter 501 directly, or through interaction with operating-system 500. OEM configuration
35 application 530 may be used during the initial system set-up, to supply gatekeeper configuration
table 506 with preselected information, thus determining the manner in which gatekeeper 504

1 will respond to certain requests from application programs 510, 512, 514, as well as application
manager 509. Both tables 502 and 506 may include pertinent credentials, including, without
limitation, process ID (PID), window ID (WID), priority, revoked and repudiated credentials,
authentication token or key, MPDA authorization, descriptive text, program status, system status,
5 accessible and excluded display regions, and the like.

Figure 6 illustrates an embodiment of the present invention, as an example, in the context
of the Microsoft WINDOWS NT® 4.0 operating system 600. (NT® is a trademark of Microsoft
Corp., Redmond, WA). Typically, application programs 610, 612, 614 interact with operating
system 600 which includes a user level 616 and kernel level 618, to transmit display requests to
10 the graphics display by (not shown) through hardware layer 625. For example, a request by a
application program 612 to provide a graphical display can be passed to WIN32 subsystem 609,
which then passes the request to the executive services function 605 within kernel level 618. The
graphical display request then is processed by WIN32K.SYS 607, with the graphical information
being transformed by graphics device drivers 603 into a format appropriate for the ultimate
15 display device.

In this particular embodiment, application programs 610, 612 and 614 request
authorization to use a MPDA from gatekeeper 604 which requests may be granted or denied
depending upon gatekeeper configuration information. Contact with the gatekeeper 604 can be
implemented by way of WIN32 subsystem 620. In this embodiment, arbiter 601 may be
20 implemented within the operating system 600 in kernel mode 608 within the context of a
windows and graphics components of WIN32K.SYS 617 and graphics device drivers 603. In
one embodiment of this configuration, arbiter 601 allows selected ones of application program
610, 612, 614 to pass graphical data to graphics device drivers 603 according to rules established
within arbiter 601. Graphics device driver 603, in turn, transmits graphic data to hardware layer
25 625 for perceptible visual display. Arbiter 601 and gatekeeper 604 may be used in conjunction
with respective access control and configuration tables (now shown), as illustrated in FIGS 1
through 5. Also, gatekeeper 604 can reside in a device physically separate from the one in which
arbiter 601 resides allowing for remote configuration by network supervisors and the like.

In Figure 7, another embodiment of the display interface according to the present
30 invention is illustrated in the context of Windows® NT® 4.0 operating system 700. As with the
example in FIG. 6, application programs 710, 712, 714 can interact with operating system 700
which includes a user level 716 and kernel level 718, to transmit display requests to the graphics
display by (not shown) through hardware layer 725. For example, a request by a application
program 712 to provide a graphical display can be passed to WIN32 subsystem 709 at user level
35 716, which then passes the request to the executive services function 705 at kernel level 718.
The graphical display request then is processed by WIN32K.SYS 707, with the graphical

1 information being transformed by graphics device drivers 703 into a format appropriate for the ultimate display device.

5 In Figure 7, the arbiter 701 and, if desired, arbiter look-up table 702, are implemented in hardware layer 725. Similar to gatekeeper 604 in Figure 6, gatekeeper 704 in Figure 7 can select among application program 710, 712 and 714 to have access to arbiter 701 and thus, an MPDA, by selected configuration parameters, which may stored in gatekeeper configuration table 706. Both tables 702 and 706 can include pertinent credentials, including, without limitation, process ID (PID), window ID (WID), priority, revoked and repudiated credentials, authentication token or key, MPDA authorization, descriptive text, program status, system status, accessible and excluded display regions, and the like. Arbiter 701 can reside on a device physically separate from gatekeeper 704, and thus those application programs which ultimately receive a MPDA can be controlled and configured by, for example, a network supervisor or monitor on a remote system. Although the Windows NT 4.0 operating-system is used to illustrate the aforementioned principles, it is to be understood that the invention herein is not limited to being implemented in a computer operating-system, but also can be implemented functionally using, for example, ASICs and other custom logic circuits, whether discrete or embedded in other functional devices.

15 Figure 8 illustrates multiple processes (e.g., process "MIKE" and process "JOHN") having access to, and creating subwindows, frames, or panes which display process-related information in main display window 800, using one or more MPDA, provided such processes have been authorized by the arbiter to access that region of the display with a MPDA. In the example of FIG. 8, the gatekeeper (not shown) can be configured to grant processes "MIKE" and "JOHN" access to the arbiter (also not shown) for the purpose of requesting a MPDA operable on a subwindow, or predetermined region of main display window 800. All other process in this example are denied access to the arbiter and, thus, to a subwindow having a MPDA. With other processes blocked from access to the arbiter, processes "MIKE" and "JOHN" each may request a MPDA for subwindows 805, 810, 815 from the arbiter. Absent a request from process "MIKE," it is possible for process "JOHN" to be given control of, along with the right to use a MPDA, for each of subwindows 805, 810, 815. In the event processes "MIKE" and "JOHN" each simultaneously request a MPDA for the same subwindow or region of display 800, process "MIKE" is given priority to the MPDA, over process JOHN, as previously determined by the gatekeeper (not shown), responsive to rules programmed in the arbiter, (not shown) or data provided in a arbiter access control table (not shown). According to the exemplary rules, if process "MIKE" is in possession of the MPDA for subwindow #1 805, process "JOHN," having lower priority, will be denied access, until process "MIKE" has terminated, and relinquished the subwindow #1 MPDA that it possesses. In the instance where process "JOHN" controls a MPDA for each of subwindows 805, 810, 815, control of the MPDA for subwindow #1 805 is

1 revoked by the arbiter, when use of subwindow #1 and control of the MPDA, is requested by,
and granted to, process "MIKE," having higher priority. Process "MIKE" then is permitted to
provide a persistent display in subwindow #1 805, regardless of the display information provided
by other processes to main display window 800, due to its dominant and exclusive control of an
5 MPDA for subwindow #1 805.

FIG. 9 illustrates another embodiment of the present invention as may be found in an
advanced graphics system 900, such as the BCM7014 Advanced 2D/3D TV Graphics System by
Broadcom Corp., Irvine, CA, as well as similar devices. System 900 can be a highly-integrated,
high-performance system for advanced graphics, text, analog video, digital video, animation, and
10 audio applications advantageous for use with analog and digital set-top box, Digital TV, and
television web browsing applications.

In this example, arbiter functionality can be implemented in Display Controller 950,
Intelligent DMA Engine 955, Accelerator 960, or CPU 970, by providing respective arbiters
901a, 901b, 901c, or 901d. In addition, arbiter access control table 902 may be implemented in
15 SDRAM 980 or system RAM 985. Furthermore, where it is desirable to include, for example,
arbiter access pre-qualification by programs and processes, gatekeeper 904 can be implemented
in CPU 970. Moreover, gatekeeper configuration table 906 can be implemented within system
RAM 985. Similar to the functionality described with respect to FIGS. 1, 2, and 4, arbiter 901a,
901b, 901c, or 901d, is embodied in hardware, although, of course, it also may be realized in
20 software, firmware, or combinations thereof.

Application 910 request access to display 920 by passing the requests through CPU 978,
which request is then processed by display system 900. Where arbiter 901a is disposed in display
controller 950 request for access is transferred to display controller 950, where arbiter 901a
determines whether process 910 will be permitted to access display 920. Display system 900 can
25 be coupled with a myriad of processing systems, including digital signal processing systems,
which through CPU 970, can provide a suitable display input to display system 900.

It is important to note that when granted the MPDA, the dominant window is unable to
be obscured by any other window or object by which the device represents data, and that the data
represented is not confined to visually-oriented data but may include aural, tactile, or any
30 perceptible data, alone or in combination. Furthermore, the inventive aspects of the present
invention can be implemented on any device representing data, including traditional computers,
pads, and tablets, wireless/cellular phones, handheld remote control device, handheld computing
device, and the like, whether stand alone, networked, or web-based, and regardless of the
medium by which data is transmitted transferred.

35 Many alterations and modifications may be made by those having ordinary skill in the art
without departing from the spirit and scope of the invention. Therefore, it must be understood

1 that the illustrated embodiments have been set forth only for the purposes of example, and that
it should not be taken as limiting the invention as defined by the following claims. A skilled
artisan would realize that the embodiments of the present invention can be realized in hardware,
in software, or in advantageous combinations thereof, and thus are within the scope of the
5 invention. The following claims are, therefore, to be read to include not only the combination
of elements which are literally set forth but all equivalent elements for performing substantially
the same function in substantially the same way to obtain substantially the same result. The
claims are thus to be understood to include what is specifically illustrated and described above,
what is conceptually equivalent, and also what incorporates the essential idea of the invention.

10

15

20

25

30

35

1 WHAT IS CLAIMED IS:

1. An apparatus for producing a perceptible representation of data, comprising:
an arbiter
 - 5 (a) selecting a dominant program from among a plurality of programs seeking a master persistence attribute to display data of the program according to a predetermined priority technique, and
 - (b) assigning the master persistence attribute to the dominant program.
- 10 2. The apparatus of claim 1, further comprising:
an access control table, coupled with the arbiter, containing indicia representative of the predetermined priority scheme.
- 15 3. The apparatus of claim 2, further comprising:
a configuration application program, coupled with the access control table, configuring the arbiter with the predetermined priority scheme.
- 20 4. The apparatus of claim 1, further comprising:
an I/O manager, coupled with the arbiter, communicating the display data between an application program and a display.
- 25 5. The apparatus of claim 4, further comprising:
a graphics device driver, coupled with the I/O manager and the display, to transmit the display data to the display.
- 30 6. The apparatus of claim 1, further comprising:
a graphics device driver, coupled with the arbiter, to transmit the display data to the display.
- 35 7. The apparatus of claim 2, wherein the indicia include one of process ID (PID), window ID (WID), priority, revoked and repudiated credentials, authentication token or key, master persistence attribute authorization, descriptive text, program status, system status, an accessible display region, and an excluded display region.

1 8. The apparatus of claim 1, wherein the arbiter further comprises one of a rules
engine, a state machine, and a content-addressable memory that provides the predetermined
priority scheme for determining dominant program priority.

5 9. The apparatus of claim 6, further comprising a gatekeeper determining selected
ones of the plurality of programs to be granted access to the arbiter to receive a master
persistence attribute according to a predetermined access scheme.

10 10. A graphic display apparatus, comprising:
a gatekeeper determining selected ones of a plurality of programs to be granted
a key to request a persistence attribute according to a predetermined access scheme.

15 11. The graphic display apparatus of claim 10, further comprising:
a graphics device driver, coupled with the gatekeeper, that couples display data
of the selected ones with a display.

20 12. The graphic display apparatus of claim 11, further comprising:
an arbiter
(a) selecting a dominant application program from among the selected ones
seeking the master display persistence attribute, and
(b) assigning the master persistence display attribute to the dominant program.

25 13. The graphic display apparatus of claim 12, further comprising:
an access control table, coupled with the arbiter, storing indicia representative of
the predetermined priority scheme.

30 14. The graphic display apparatus of claim 10, further comprising:
an I/O manager, coupled with the gatekeeper, managing graphical data between
the selected ones and a display.

35 15. The graphic display apparatus of claim 10, further comprising:
an application manager, coupled with the gatekeeper, preventing unauthorized
access to an operating system by the selected ones.

- 1 16. The graphic display apparatus of claim 15, further comprising:
 a graphics device driver, coupled with the application manager, to transmit
graphical data to display data on the display.
- 5 17. The graphic display apparatus of claim 10, further comprising:
 a configuration application program, coupled with the gatekeeper, configuring the
gatekeeper with the predetermined priority scheme.
- 10 18. The graphic display apparatus of claim 10, further comprising:
 a configuration table, coupled with the gatekeeper, storing an indicia
representative of the predetermined priority scheme.
- 15 19. The apparatus of claim 18, wherein the indicia include one of process ID (PID),
window ID (WID), priority, revoked and repudiated credentials, authentication token or key,
master persistence attribute authorization, descriptive text, program status, system status, an
accessible display region, and an excluded display region.
- 20 20. A graphic display apparatus comprising:
 (a) a gatekeeper determining selected ones of a plurality of program to be
granted a master persistence display attribute according to a predetermined access technique, and
 (b) an arbiter;
 (1) selecting a dominant program from among the selected ones
seeking the master persistence display attribute, and
 (2) assigning the master persistence attribute to the dominant program
25 according to a predetermined priority technique.
- 30 21. The graphic display apparatus of claim 20, further comprising:
 one of a configuration table, coupled with at least one of the arbiter and the
gatekeeper, containing first indicia representative of a predetermined priority scheme, and an
access control table, coupled with at least one of the arbiter and the gatekeeper, containing
35 second indicia representative of a predetermined priority scheme.
22. The graphic display apparatus of claim 20, further comprising:
 a configuration application, coupled with at least one of the configuration table
and the access control table, configuring at least one of the arbiter and the gatekeeper.

1 23. The graphic display apparatus of claim 19, further comprising:
 an I/O manager, coupled with at least one of the arbiter and the gatekeeper, for
communicating the display data between an application program and a display.

5 24. The graphic display apparatus of claim 23, further comprising:
 a graphics device driver, coupled with the I/O manager and the display, that
transfers the display data to the display.

10 25. The graphic display apparatus of claim 24, further comprising:
 a display buffer coupled with the graphic display driver.

 26. The graphic display apparatus of claim 20, further comprising:
 a graphics device driver, coupled with at least one of the arbiter and the
gatekeeper, that transfers display data to a display.

15 27. The graphic display apparatus of claim 26, further comprising:
 a display buffer coupled with the graphic display driver.

 28. The graphic display apparatus of claim 26, further comprising:
20 an I/O manager, coupled with the graphic display driver, communicating between
an application program and the display.

 29. The graphic display apparatus of claim 20 further comprising:
 an application manager, coupled with at least one of the gatekeeper and the
25 arbiter, preventing unauthorized access to an operating system by a program.

 30. The apparatus of claim 21, wherein at least one of the first indicia and the second
indicia include one of process ID (PID), window ID (WID), priority, revoked and repudiated
30 credentials, authentication token or key, master persistence attribute authorization, descriptive
text, program status, system status, an accessible display region, and an excluded display region.

 31. A graphics system, comprising:

- a. a video input receiving a graphical data signal;
 b. a video output coupled with a display;
35 c. a display controller coupled with the video input signal and selectively
transmitting the graphical data signal to the video output; and

1 d. an arbiter coupled with the display controller, the arbiter effecting the
selectively transmitting by granting a persistence attribute according to a
predetermined priority scheme, the display controller selectively
transmitting responsive to the arbiter.

5 32. The graphics system of claim 31, further comprising a CPU interface for coupling
the graphics system to a CPU, the CPU receiving display control signals and the arbiter being
responsive thereto.

10 33. The graphics system of claim 32, wherein the CPU includes a gatekeeper, the
gatekeeper operably coupled with the arbiter and transmitting the predetermined priority scheme
thereto.

15 34. The graphics system of claim 32, wherein the CPU includes a gatekeeper, the
gatekeeper operably coupled with the arbiter and selecting display control signals having access
to the arbiter.

20 35. The graphics system of claim 34, further comprising an arbiter access control table
configured to receive indicia relevant to the priority scheme.

25 36. The graphics system of claim 35, wherein the indicia include one of process ID
(PID), window ID (WID), priority, revoked and repudiated credentials, authentication token or
key, master persistence attribute authorization, descriptive text, program status, system status,
an accessible display region, and an excluded display region.

30 37. A method of assigning a persistence attribute to at least one of a plurality of
dominant programs, comprising:

- (a) requesting the master persistence attribute from a gatekeeper;
- (b) assigning a set of priority rules to the gatekeeper via a configuration
application program;
- (c) the gatekeeper granting keys to selected dominant application programs
allowing access to an arbiter;
- (d) the arbiter examining an arbiter access control table storing the
predetermined priority scheme; and
- (e) the arbiter assigning the persistence attribute to the at least one of a
plurality of dominant application programs granting access to a display window.

1 38. A computer program product recorded on a computer readable medium for assigning a master persistence attribute to at least one of a plurality of programs, comprising:

5 (a) computer readable program code by which a gatekeeper grants an access token to selected programs allowing access to an arbiter according to a predetermined access scheme; and

 (c) computer readable program code by which the arbiter assigns the master persistence attribute to a dominant one of the selected programs thereby granting access to a preselected display window.

10 39. The computer program product of claim 38, further comprising computer readable program code by which the arbiter examines an arbiter access control table storing the predetermined priority scheme.

15 40. The computer program product of claim 39, further comprising computer readable program code which assigns a set of access rules to the gatekeeper and assigns a set of priority rules to the arbiter, via a configuration application program.

 41. A method of assigning a master persistence display attribute to at least one of a plurality of dominant application programs comprising:

20 (a) requesting the persistence attribute from a gatekeeper;
 (b) the gatekeeper accessing a configuration table storing a predetermined priority scheme;

 (c) the gatekeeper granting keys to selected dominant application programs;
 (d) the selected dominant application programs applying the keys to access
25 an arbiter which examines an arbiter access control table storing a predetermined priority scheme; and

 (e) the arbiter assigning the master persistence display attribute to the at least one of a plurality of dominant application programs granting access to a display window.

30 42. An apparatus for producing a perceptible representation of data, comprising an arbiter that selects a dominant program from among a plurality of programs seeking a master persistence attribute to display data of the program according to a predetermined priority technique, and assigns the master persistence attribute to the dominant program, wherein the perceptible representation of data is rendered on one of a computer, a communication pad, a
35 communication pad, a telephony device, a handheld remote control device, and a handheld computing device.

1 43. The apparatus of claim 42 wherein the medium by which data is communicated to
the apparatus includes one of a wireless/RF channel, a wire-based channel, a cable-based
channel, and a fiberoptic channel.

5

10

15

20

25

30

35

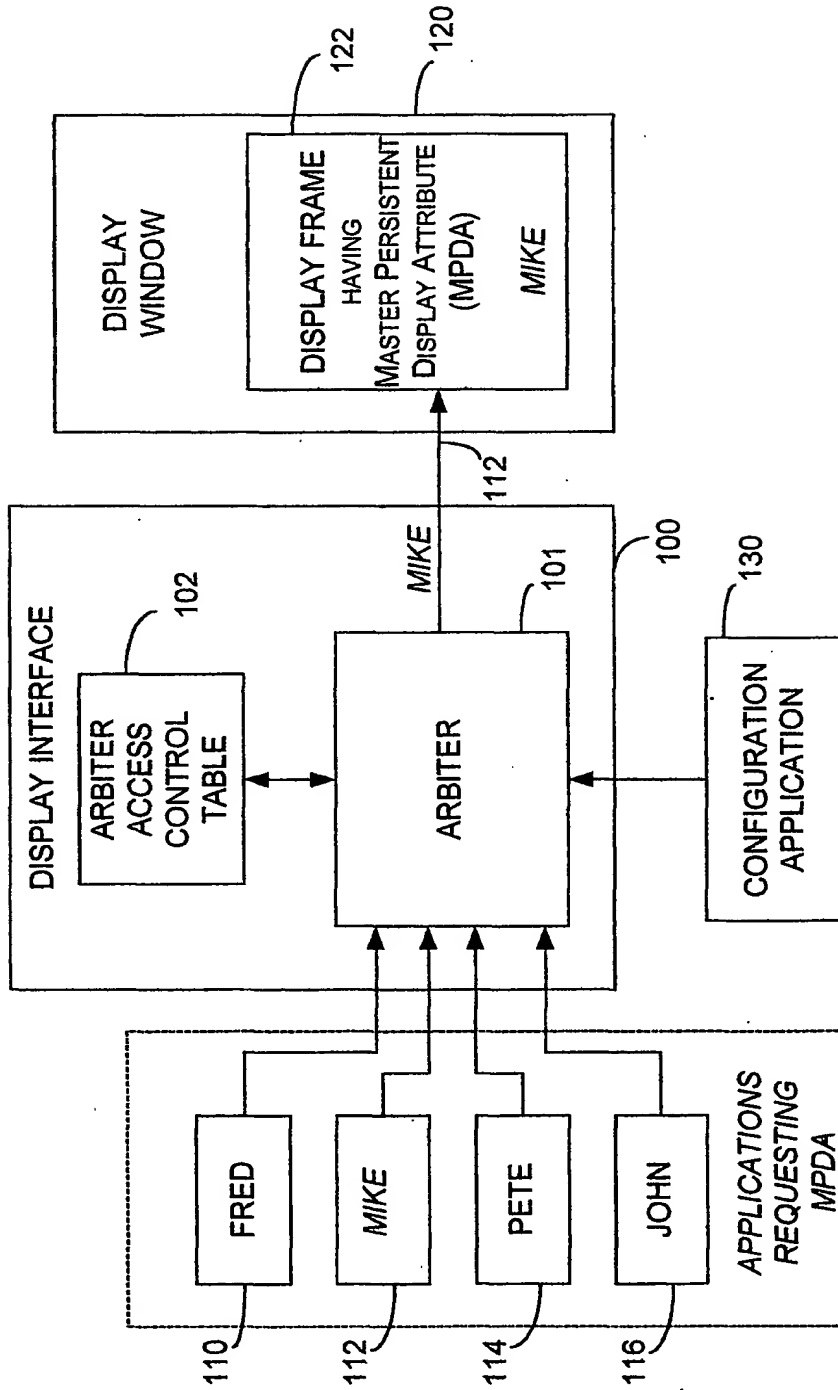


FIG. 1

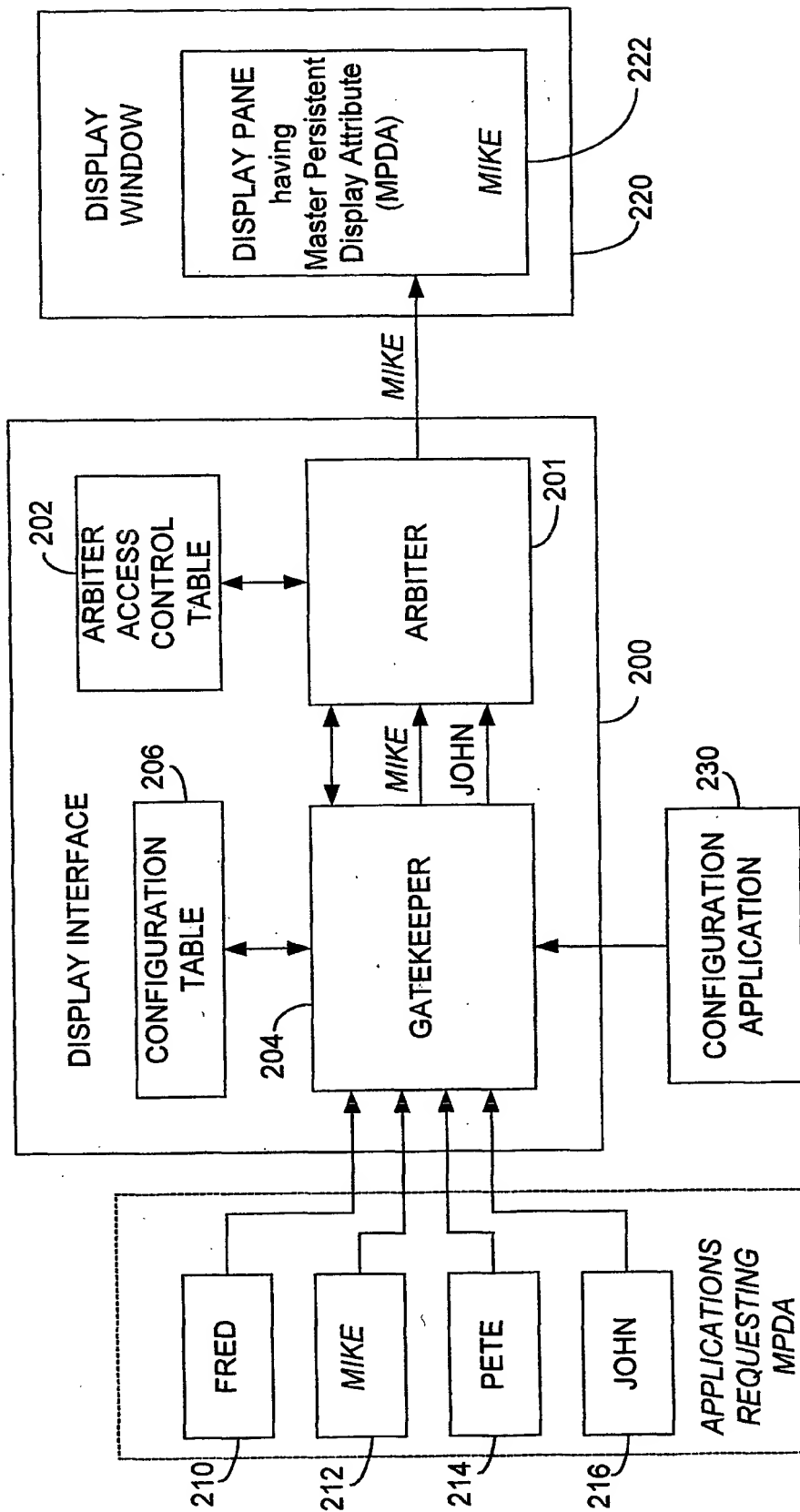


FIG. 2

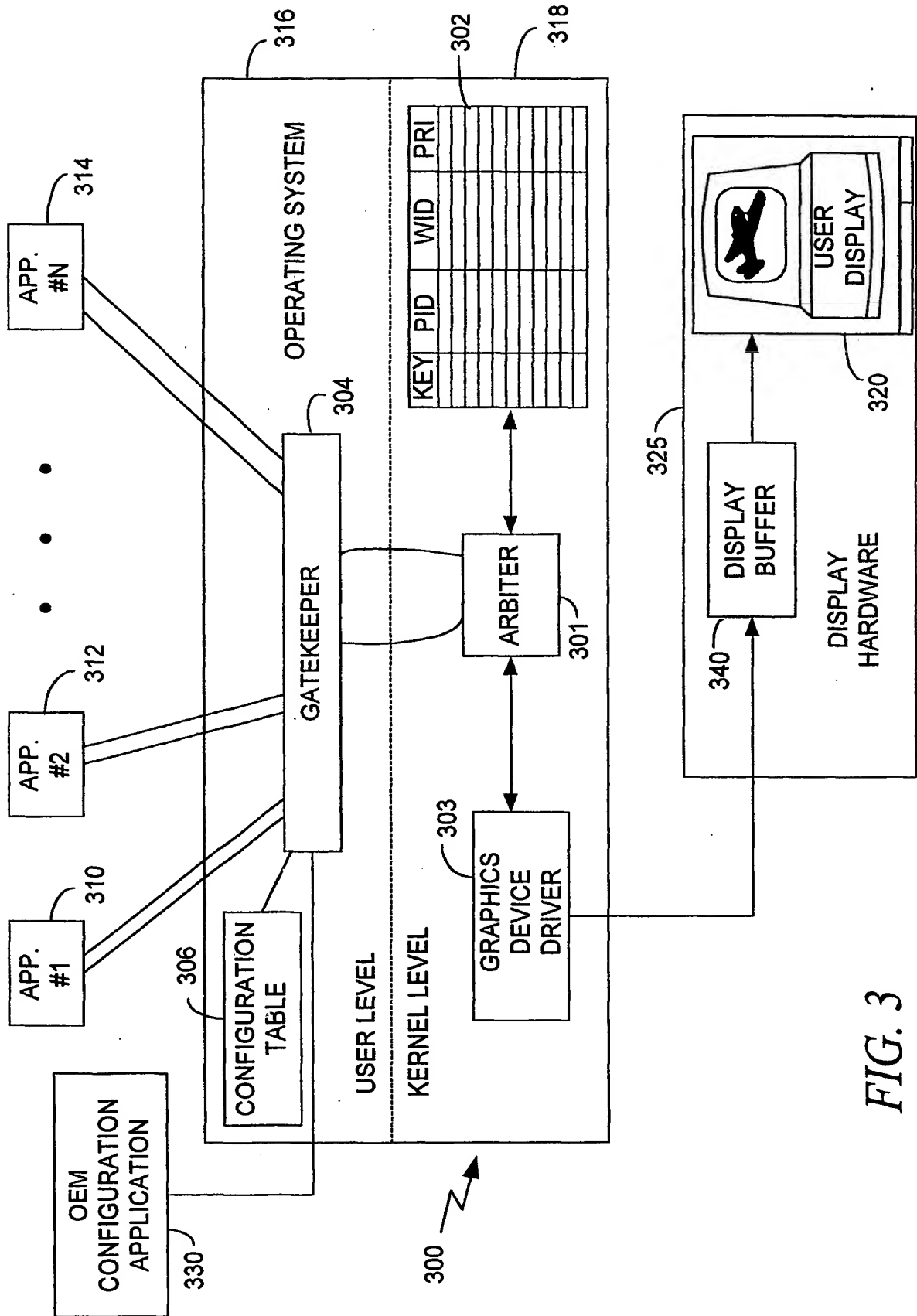


FIG. 3

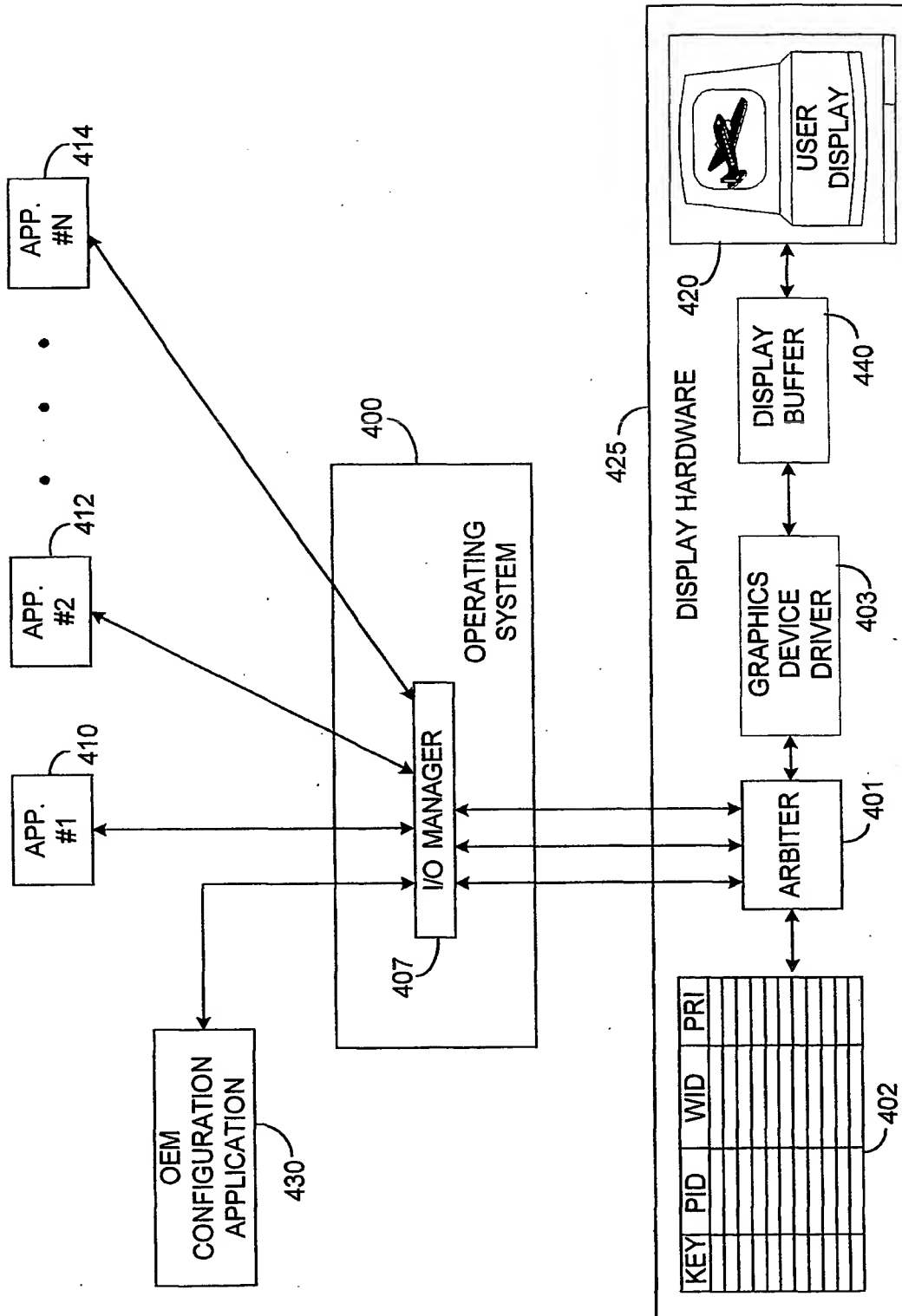


FIG. 4

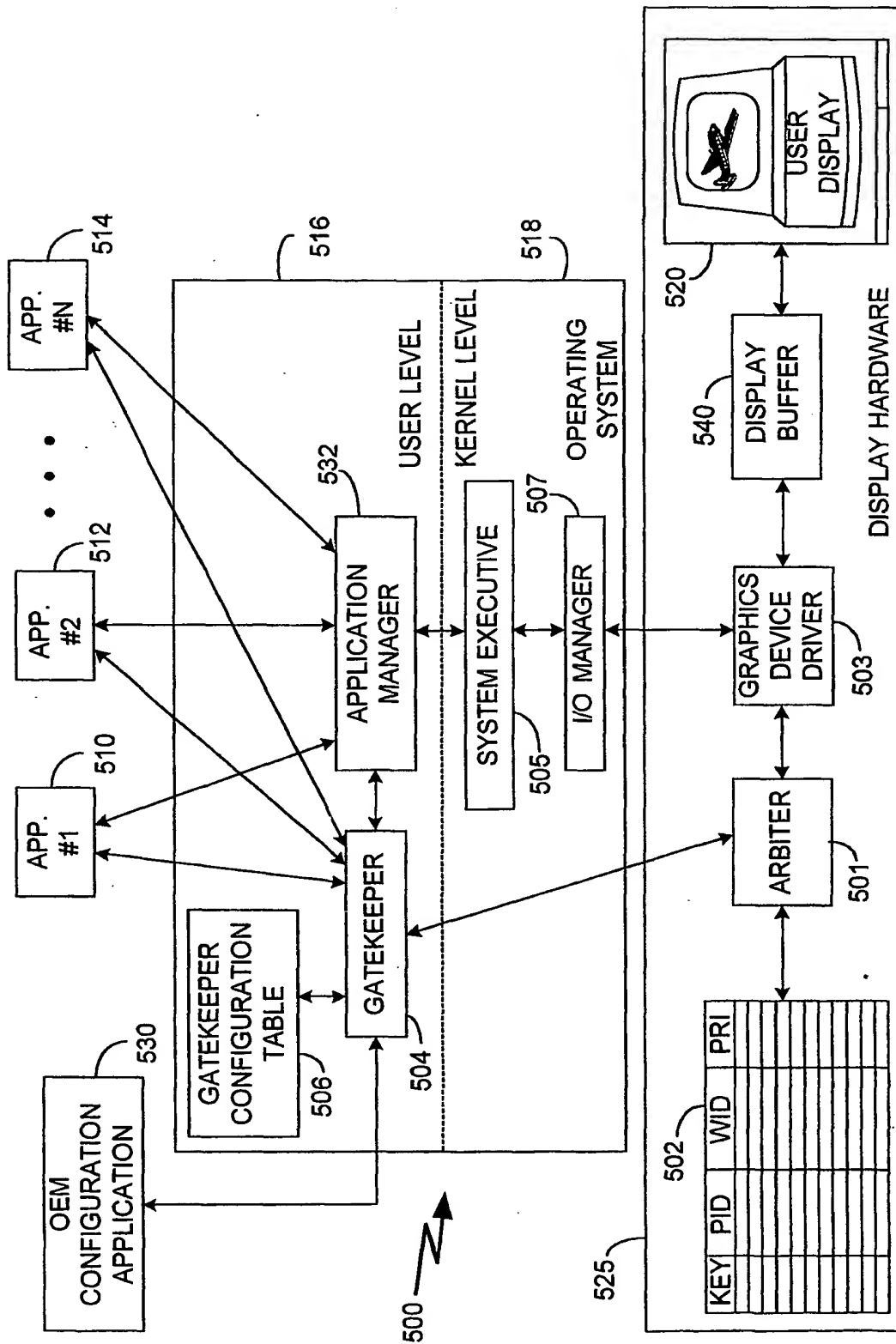


FIG. 5

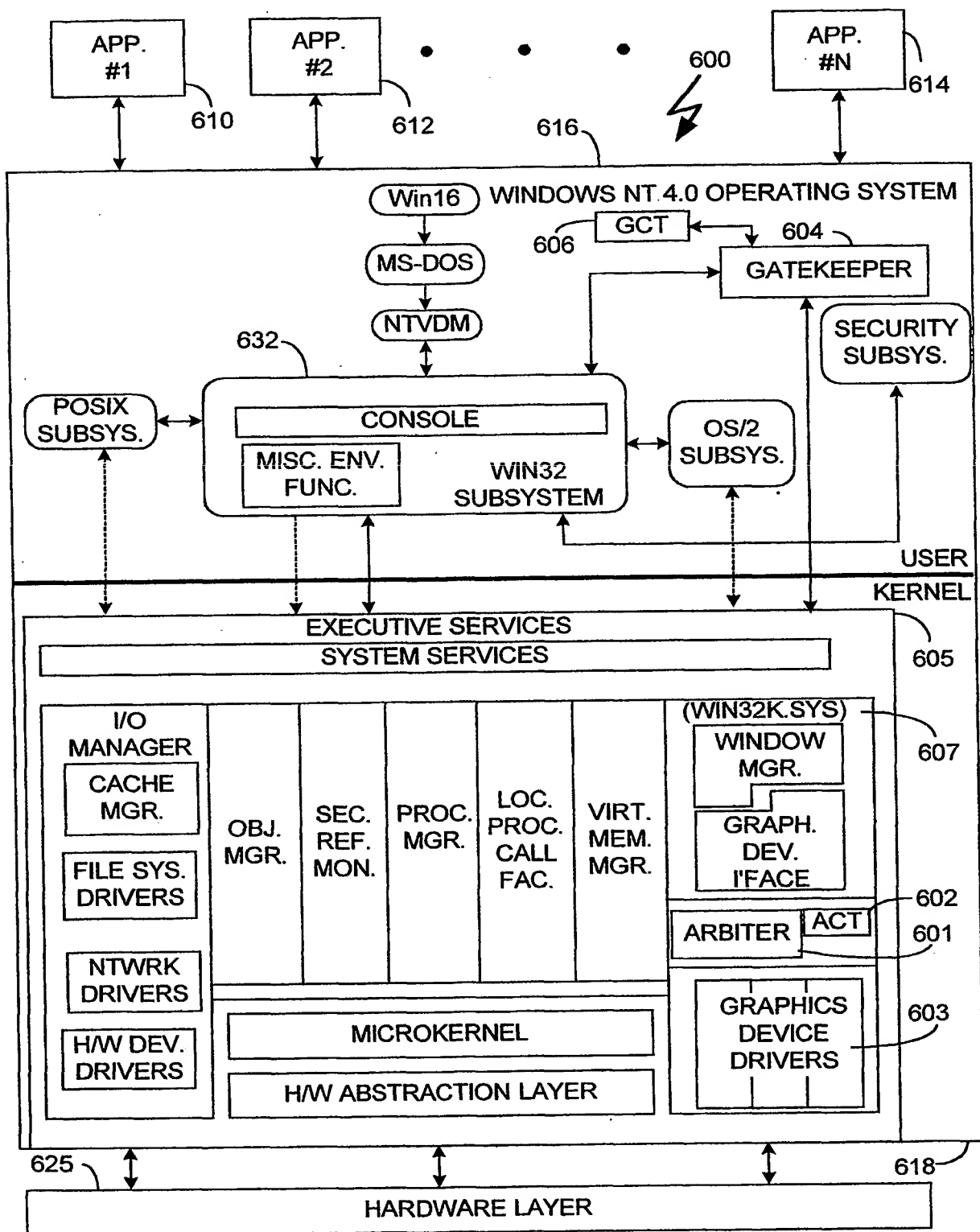


FIG. 6

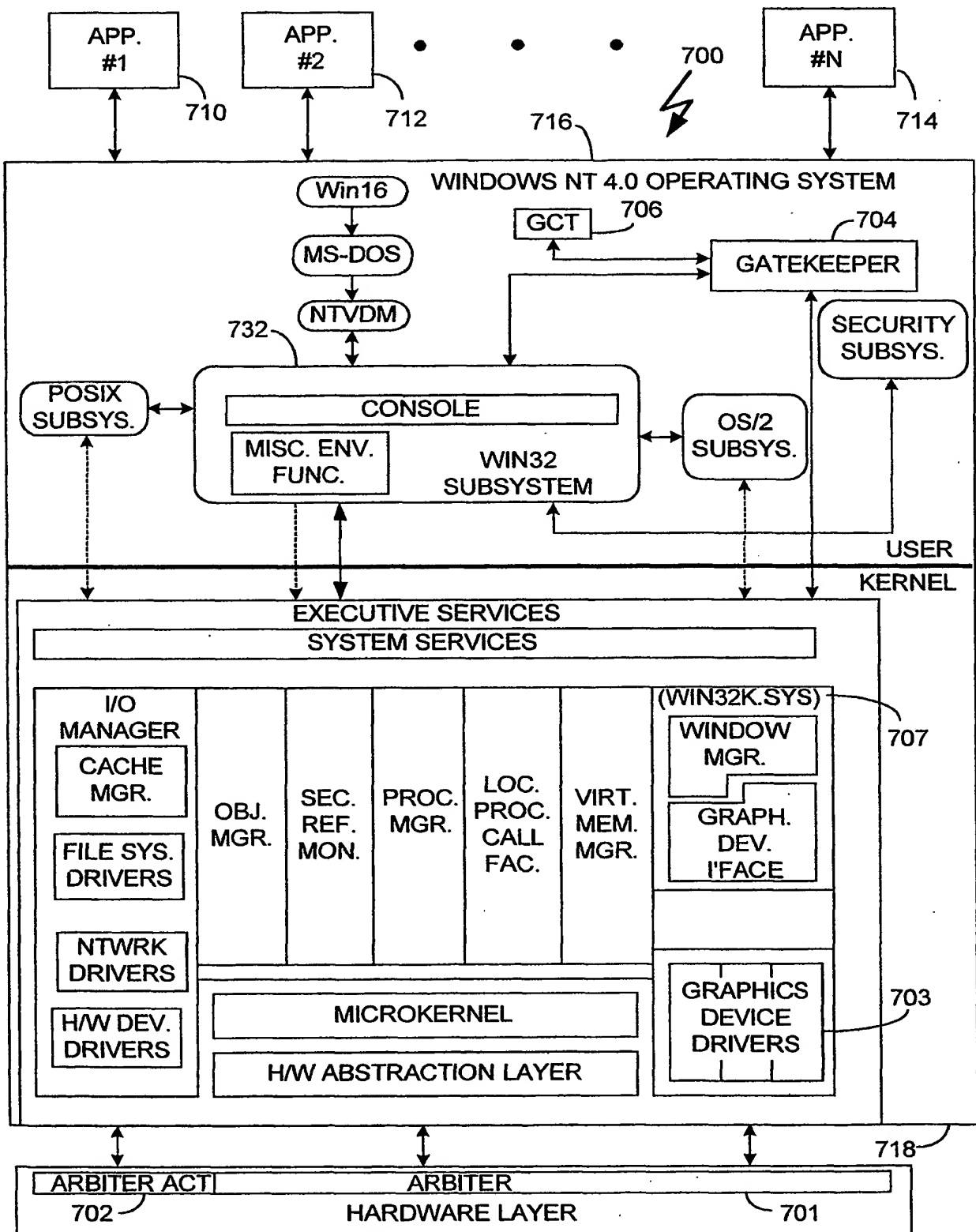
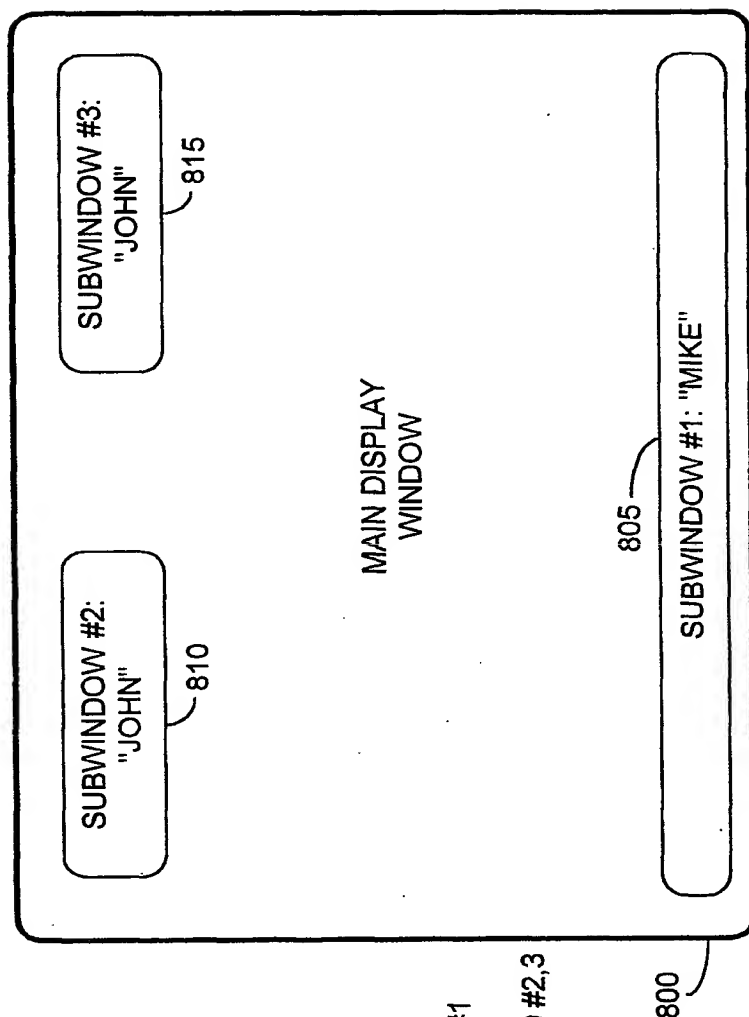


FIG. 7



GATEKEEPER CONFIGURATION

1. Accept any request from process "MIKE"
2. Accept any request from process "JOHN"
3. Deny access to all other processes
4. If conflict, give "MIKE" priority over "JOHN"

ARBITER CONFIGURATION

1. Select "MIKE" for access to subwindow ID #1
2. Deny "JOHN" access to subwindow ID #1
3. Select "JOHN" for access to subwindows ID #2,3

FIG. 8

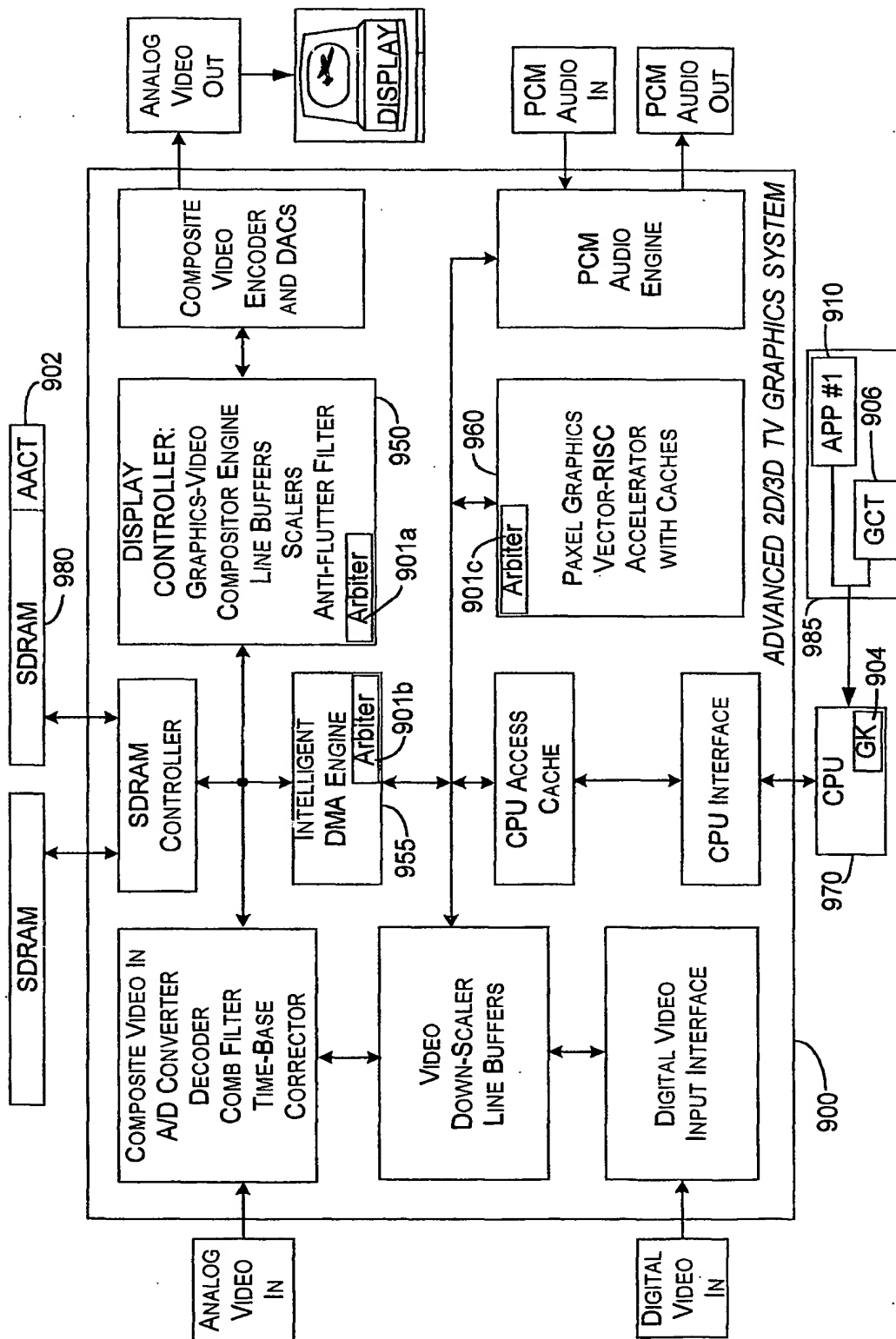


FIG. 9

INTERNATIONAL SEARCH REPORT

tional Application No

PCT/US 01/12795

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G09G5/14 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G09G G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 585 191 A (IBM) 2 March 1994 (1994-03-02) column 8, line 58 -column 11, line 2 ----	1-7, 42, 43
A	US 5 640 498 A (CHEW CHEE HENG) 17 June 1997 (1997-06-17) column 3, line 10 -column 4, line 27 column 6, line 4 -column 9, line 24 column 15, line 24 - line 33 ----	1, 2, 7
A	US 6 005 575 A (COLLERAN JOHN D ET AL) 21 December 1999 (1999-12-21) column 5, line 13 -column 9, line 31 ----	1
A	US 6 031 530 A (TRUEBLOOD JOHN WARREN) 29 February 2000 (2000-02-29) -----	

☐ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *G* document member of the same patent family

Date of the actual completion of the international search

20 July 2001

Date of mailing of the international search report

27/07/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Amian, D

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 01/12795

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0585191 A	02-03-1994	JP 6301554 A	28-10-1994
US 5640498 A	17-06-1997	US 5678034 A	14-10-1997
US 6005575 A	21-12-1999	NONE	
US 6031530 A	29-02-2000	US 5675755 A	07-10-1997
		CA 2177801 A	08-12-1996
		EP 0747805 A	11-12-1996
		JP 9146751 A	06-06-1997

This Page Blank (uspto)